An Empirical Study of Speed and Communication in Globally Distributed Software Development

James D. Herbsleb and Audris Mockus

Abstract—Global software development is rapidly becoming the norm for technology companies. Previous qualitative research suggests that distributed development may increase development cycle time for individual work items (modification requests). We use both data from the source code change management system and survey data to model the extent of delay in a distributed software development organization and explore several possible mechanisms for this delay. One key finding is that distributed work items appear to take about two and one-half times as long to complete as similar items where all the work is colocated. The data strongly suggest a mechanism for the delay, i.e., that distributed work items involve more people than comparable same-site work items, and the number of people involved is strongly related to the calendar time to complete a work item. We replicate the analysis of change data in a different organization with a different product and different sites and confirm our main findings. We also report survey results showing differences between same-site and distributed social networks, testing several hypotheses about characteristics of distributed social networks that may be related to delay. We discuss implications of our findings for practices and collaboration technology that have the potential for dramatically speeding distributed software development.

Index Terms—Global development, collaboration, delay, speed, awareness, informal communication.

1 Introduction

Communication and coordination issues in large software engineering projects have always been formidable (e.g., [5], [9]). Increasingly, engineers and managers must add the challenges of coordinating work across sites, spanning national, language, and cultural barriers (see, e.g., [6]). Driven by market and resource requirements, the push toward globalization has generated a wide variety of problems for software developers [24].

Previous research [21], [23] suggests that cross-site communication and coordination issues cause a substantial loss of development speed. In this paper, we investigate relationships among delay, communication, coordination, and geographic distribution of work, in order to shed light on the possible mechanisms responsible for introducing delay. Further, we discuss how the understanding achieved from this empirical research informs the development of tools and practices to improve distributed development.

In the remainder of this introduction, we review literature on geographically distributed engineering work and how it differs from colocated work and conclude with our specific research questions. In the following section, we describe our empirical methods. Then, we present new results on communication patterns across and within sites and results showing the relationship of distributed work, delay, and other important variables. Finally, we draw out the implications of these observations for achieving success in distributed work and conclude the paper.

1.1 Communication and Distance

In sharp contrast to the popular image of software developers as relatively introverted and isolated, they, in fact, spend a large proportion of their time communicating. For example, in an empirical study of time use by developers in a large software engineering organization, Perry et al. [39, p. 41] reported that "one of the most salient impressions conveyed by observation was the sheer amount of time each developer spent in informal communication." The developers in their study spent an average of 75 minutes each day in "unplanned interpersonal interaction."

In an eight-month study of a medium-sized telecommunications software project [22], an analysis of time sheets indicated that about 50 percent of time was spent in "group work" (meetings and unplanned work-related discussions) during the first month, and this level dropped fairly steadily until only about 10 percent of time was spent in group work during the last month. Design activities, in particular, seemed to require a very large proportion of collaborative work.

In contrast to the frequent interaction of colocated work, there is very convincing evidence that the frequency of communication generally drops off sharply with physical separation among coworkers' offices and that the sphere of frequent communication is surprisingly small. T. Allen [3], in a study of engineering organizations, reported that, when engineers' offices were about 30 meters or more apart, the frequency of communication dropped to nearly the same

- *J.D. Herbsleb is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891. E-mail: jherbsleb@acm.org.*
- *A. Mockus is with Avaya Labs Research, 233 Mount Airy Road, Basking Ridge, NJ 07920. E-mail: audris@avaya.com.*

Manuscript received 22 Oct. 2001; revised 10 July 2002; accepted 7 Nov. 2002.
Recommended for acceptance by K. El Emam.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 115225.

low level as people with offices separated by many miles. Kraut et al. [27] found similar results for scientists.

These findings are particularly troubling in rapidly evolving, high technology environments, where the competitors, products, standards, and customers routinely create a demand for significant, unforeseen changes throughout the development cycle. In organizations with rapidly changing environments and "unstable" projects, informal communication is particularly important [16], [28]. For example, as requirements change, it is hard for the formal mechanisms of communication, such as specification documents, to react quickly enough. Often, news of change, its significance, and its potential impact is propagated informally and very quickly among the development staff. Under such conditions, the pattern of lateral communication across sites should be particularly important.

Research showing the importance of informal communication has lead to a variety of technologies designed to stimulate casual conversation among workers at different sites. These technologies have included video [1], [11], [15], [35], audio [25], and text [7]. To this list, we must now also add instant messaging, a technology that is beginning to infiltrate the work place (e.g., [26], [34]).

These observations about communication and distance also highlight the importance of understanding the dependencies among the various kinds of work involved in software development [17]. In a study of six software engineering organizations, Grinter et al. [18] observed four different ways of organizing work across sites that evolved within a single global corporation. Each represented an attempt to minimize requirements for cross-site communication in the context of particular types of product architectures and mechanisms for coordinating work. There are also indications, from a study of an automotive engineering group, that, where possible, engineers will try to reduce the coupling of distributed work [36].

In a case study of a software engineering organization spread across several sites, Herbsleb and Grinter [21] investigated how the organization used a number of mechanisms, including plans, processes, and interface specifications, to coordinate the distributed work. Each mechanism, however, was vulnerable to imperfect foresight and unexpected events, which required substantial communication to coordinate activities and renegotiate commitments. Despite the need for communication, there was a nearly total absence of informal, unplanned communication across sites.

The difficulties of knowing who to contact about what, of initiating contact, and of communicating effectively across sites led to a number of serious coordination problems. Among these problems were unrecognized conflicts among the assumptions made at different sites and incorrect interpretation of communications. The most frequent consequence of cross-site problems was *delay* in the resolution of work issues. By delay, we mean the additional time it takes to resolve an issue when more than one site is involved. So, for example, if a part of the design or code needs to be changed, or if someone needs a better understanding of how some part of the product works, people at more than one site may need to be involved in information exchange, negotiation, and so on, in order to find a solution. Such issues arise very frequently in software development.

Qualitative studies (e.g., [21], [18]) have shown how individuals are disrupted by cross-site coordination challenges. But, questions remain about the cumulative effects, for example, how distance affects the speed with which software engineering tasks are accomplished, and how distance is related to other important variables that influence speed, such as the size of a task, or the number of people involved. In addition to being important research questions, these are critical pragmatic issues as businesses become more globally distributed. Speed to market is a critical success factor for new products (e.g., [10]).

In this paper, we use three independent sources of data to examine the effect of distributed work on speed and then examine a number of properties of distributed versus same-site communication that may account for these differences. Finally, we discuss the implications of these findings for tools to address these communication issues.

1.2 Research Questions

This paper reports a study of two geographically distributed organizations that pays particular attention to the effects of geographic distribution on delay in the development life cycle. We also examine the patterns and quality of communication in order to shed light on possible causes of delay.

Does distributed work introduce delay, as compared to same-site work? Previous research suggests that working across sites introduces substantial delay because of reduced communication, difficulty in finding the right person and establishing contact, as well as having an effective collaborative session. We examine quantitative data comparing the time required for similar same-site and distributed software work items.

What factors influence the time interval required to make a software change? What role, if any, does spreading work across multiple sites play in lengthening this interval? Assuming that there is an association between distributed work and longer intervals, there are many distinct ways in which working across multiple sites might introduce this delay. By modeling the time interval required to make a software change, we extract evidence helpful in determining the nature of the relationship and what causal mechanisms are plausible.

What differences are there between same-site and distributed social "networks" and their effectiveness? One of the possible causes of multisite delay is the difficulty of communication and coordination inherent in distributed work. In order to begin to understand this issue, we address several basic questions about communication within a site and how it differs from communication across sites. For example, what is the relative size of local and distributed social networks? Is there a perception of greater misunderstanding of tasks, priorities, plans, and changes across sites?

To what degree is work at the different sites interdependent? Does interdependence diminish over time? This is an important question because it may be that software organizations split across sites are able to quickly rearrange their work assignments so as to minimize

interdependencies. In fact, there is some evidence that this happens on occasion [36]. If it is generally the case that problems of geographic distribution are only temporary, then one should merely look for ways of surviving this state and moving through it more quickly. On the other hand, it is often quite difficult to rearrange work assignments among sites because of the location of experts in particular disciplines and location of physical resources. If work remains highly interdependent across sites, then one either needs to find ways to make distributed work more effective, find new ways to reduce interdependence, or both. To examine trends in interdependence, we have developed a measure of the coupling of work.

2 Sites and Methods

In this section, we describe the sites of study, including some background on the products built. We also discuss how the work is divided among sites. We conclude with a description of the methods used to analyze and collect the data.

2.1 Sites

We chose two departments of the company to study. Both work in complex areas of telephony, where the market requirements and standards are changing rapidly. This makes coordinating the development work extremely difficult and subject to continuous change. In addition, their products compete in an aggressive market, which brings extreme time pressures to development work.

Department A has four locations, one in the United Kingdom (UK), one in Germany, and two in India. The number of people grew over the period of the study, but averaged about 40 at the UK site, 75 in Germany, and 30 in India. These sites exchange information frequently and make decisions that require cross-site synchronization. The German site had existed for a number of years as part of another company before being acquired by Lucent about a year before this study began. The people there had considerable experience working together on similar systems. They developed much of the software that was closely connected to the hardware (they were colocated with hardware design), participated in architecture work, and had responsibility for network test. The UK site had only existed for about three years and, thus, had no existing relationships to any other Lucent site. The UK site was colocated with product and project management and was responsible for most of the architecture work and for application software. One Indian site was also about three years old. The other was a software contractor, not actually a part of Lucent, but it had worked with the German and UK sites for several years. The contractor developed software in highly circumscribed areas deemed not within Lucent's core competencies. The Lucent site, during the period of this study, was primarily working on performance testing and enhancement. With the exception of having only restricted access to the Lucent intranet, the contractor site participated fully in projects, in ways indistinguishable from Lucent sites.

Department B has two locations that are relevant to this study, one in the Midwestern US and one in Ireland. The Midwestern site, which averaged about 30 people, has been in the company for many years and has had fairly close working relationships with other US sites, but had not previously worked with the Irish site. It designed and built software for two major components, one that was a modification of a previous product and another that was a next generation version of a similar component. The Irish site was relatively new, having existed for only about three years and averaged about 60 people. For its entire existence, it has collaborated on various projects with other sites in Europe, Asia, and the US. It was responsible for all other components, as well as most architecture work.

Department A and Department B worked on different network elements for a telecommunication system. There was no overlap in sites or personnel between the departments. The software built in the two departments was also quite different, in that Department A built software closely related to real-time processing and switching of calls, while Department B built software for network monitoring, maintenance, and administration, which was much more focused on interfaces with other network elements and human-computer interfaces.

2.2 Methods

Our results draw on modification requests from the change management system and survey data.

2.2.1 Data Generated by Change Management System

Like many software development organizations, both departments we studied used a change management (CM) system to organize and track their development work. CM systems typically provide mechanisms for versioning the code and some ability to manage concurrent changes in a structured way. CM systems track development work by correlating the actual changes in the code with requests to make those alterations. In the development organizations under study, the basic tracking unit is called the Modification Request (MR), which is a request to incorporate a specific functionality into the software. Some MRs ask for new functionality, others ask for specific problems to be fixed. All development work in both organizations was done within the framework of an MR, using Sablime™ to track changes and ClearCase™ to provide version control.

Modification requests are generated either when problems are identified or when new features are requested. These MRs are reviewed by a change control board (CCB), which determines which requests will be accepted, the priority of accepted requests, and the most appropriate person to whom to assign the request (whom we will call the MR owner). If the MR owner cannot complete the MR on their own because they do not have needed expertise or authority to make all of the needed change, the owner is responsible for identifying and recruiting appropriate staff.

The software used for tracking MRs automatically collects several valuable types of data. It establishes a record for each MR of who made the request, the date the request was made (or "opened"), and each change ("delta") that is made to the code base in order to fulfill the request. For each delta, it records the login of the person submitting the code and the time, size, and date of the submission. Large, complex MRs typically have many deltas, whereas small, simpler MRs have only a few, or even just one delta.

MRs are the basic unit of work in this software development. Moreover, MRs and their equivalents in other CM systems are pervasive in most software development work. By performing simple calculations on the MR data, it is possible to derive several important measures (e.g., [33]), described below.

2.2.2 Survey

We conducted two surveys, both in Department A. (Unfortunately, we were not able to survey Department B because of organizational changes.) In November 1998, 117 employees located in Germany and the UK were invited to complete a Web-based questionnaire. Most of the workers were software engineers, with some managers and some administrative support personnel. In June 1999, a second administration of a similar survey was undertaken. In all, 160 employees in Germany, the UK, and two sites in India were invited to take this survey.

The first questionnaire[1] consisted of 68 items and the second had 65. Both included questions covering demographics, patterns of communication, working relationships, coordination, information exchange, and language. The respondents provided two answers for most questions: one with regard to local coworkers and the other with regard to distant coworkers. Many identical questions were included in both administrations of the survey. There were some deletions and additions, however, in order to drop questions that did not seem useful, to measure new variables, and to refine our measurements of others.

The surveys were administered in English in the UK and India. A German language version was produced using back translation techniques and was available for German speakers. Both versions were pilot tested with members of the organization being studied.

Overall, 98 of 117 surveyed employees completed the first questionnaire, for a response rate of 83 percent.[2] Across the four sites, 160 employees were invited to participate in the second wave survey. We obtained usable responses from 96 individuals, for a response rate of 60 percent.[3]

3 Results

3.1 Delay

We have two different measures of delay that allow us to compare single-site work with distributed work and to validate different measures against each other. One measure is derived from our second survey, which included the following two questions:

- How many times in the past month was your own work delayed because you needed information, discussion, or a decision from someone at your site or another site?
- What was the average length of the delays you experienced before acquiring the needed information, having the discussion, or being informed of the decision by the person from your site or the other site?

For each question, the respondent answered by supplying one number for "local site" and another number for "distant site." Of the 92 respondents, 39 reported at least one delay in the past month for the local site and 48 reported at least one delay for the remote site. Averaged over all 92 respondents, the mean number of local delays was 2.1 delays per month and the mean duration was .9 days. For cross-site delays, the mean number was 1.9 delays per month and the mean duration was 2.4 days.

In order to test the significance of the differences in number and duration of local and remote delay, a paired observation t-test was performed on a square root transformation of the data.[4] The difference between the number of delays (local versus remote) was not significant ($t = 0.1758$, $df = 91$, not significant). The difference, however, in duration (local delays versus remote delays)-was statistically significant ($t = 2.5079$, $df = 91$, $p < 0.02$). In summary, while there is no significant difference in the number of delays reported, their duration does vary significantly with delays crossing sites taking almost a day and a half longer than single site cases.

We see similar findings in the MR data. We extracted all of the "single-site" MRs, i.e., where everyone involved in the MR (the person who made the request and all the people who carried out the work of making the change) resided at one site, and compared them with the "distributed" MRs, which involved at least two sites. The average single-site MR in Department A took about five days to complete, from the time the work began (first delta) until the last change was made (last delta). (We refer to this period of time as the "MR interval"). In contrast, MRs which involved more than one site took 12.7 days, more than 2.5 times as long, to complete. The difference is statistically significant ($p < 0.001$) using a t-test.

The story is remarkably similar in Department B, except that overall MR intervals were longer, and the absolute differences between single-site and distributed MRs were greater. The average MR interval for a single-site MR was about seven days. In contrast, MRs that involved more than one site took 18 days to complete. Again, we see an MR interval for distributed MRs that is approximately 2.5 times the single-site interval. Again, the difference is statistically significant ($p < 0.001$) using a t-test.

3.2 Modeling MR Interval

To understand potential mechanisms by which distributed work might introduce delays, we used statistical modeling techniques to build a model of the MR interval. We began by using only data from Department A, reserving data from Department B for the purposes of replication. None of the following analyses were performed on Department B data

1. The questionnaires are available from the authors. Send e-mail to jherbsleb@acm.org.

2. In this first survey, 22 of the responses were from sites we have not yet been able to visit. Because we were not certain, we understood the relation of these sites to the two primary sites, these responses were eliminated from the 1998 survey.

3. Four of the respondents reported no contact with any other site, so their data were eliminated.

4. The scale for the delay data is truncated at zero, so the distribution is skewed and, consequently, not suitable for a t-test. A square root transformation on interval produced a good approximation to the normal distribution and was used in the tests.

TABLE 1
Correlations among the Variables for Department A

	interval*	number of people*	diffusion*	size*	time	severity	fix	distributed
interval*	1.00	0.34	0.52	0.49	0.19	-0.07	-0.10	0.10
number of people*		1.00	0.26	0.23	-0.01	0.13	0.02	0.29
diffusion*			1.00	0.76	0.07	-0.09	-0.15	0.07
size*				1.00	0.27	0.08	-0.14	0.05
time					1.00	-0.02	0.02	0.13
severity						1.00	0.07	0.04
fix							1.00	0.09
distributed								1.00

until the analysis of Department A data was completed and submitted for publication [23]. We followed this procedure in order to avoid the pitfalls of a post hoc analysis. In effect, we used Department A data for exploration and initial confirmation of hypotheses and Department B data for independent confirmation.

We selected a number of change measures that could be related to the MR interval (see [33]).

- *Number of people*. We expect that the MR interval may increase with the number of people involved (a person opening an MR or making a delta) in the change because of potential communication and coordination issues.
- *Diffusion*. We expect that diffuse changes spanning large parts of the system would take longer to complete than localized changes, so we chose the number of modules touched by the MR (i.e., modules containing files that were changed) as the measure of diffusion.
- *Size*. We expect that larger changes are more likely to take more time to complete, and we chose the number of delta to represent the size of the change.
- *Time*. We selected the time of the first delta as a time covariate to control for any time-related factors such as business environment and software release cycles.
- *Bug fix*. We expect that bug fixes (as estimated in [31]) might have a different interval than other MRs.
- *Severity*. We included an indicator of high severity of an MR as a measure of priority (because the priority was not recorded). We expect that high severity MRs would be resolved faster.
- *Distributed*. Finally, we expect that distributed changes (involving people from more than one site) would take longer than single-site changes.

Due to extremely skewed distributions, we performed a natural log transformation of MR interval, size, diffusion, and the number of people. The Tables 1 and 2 present the results of an analysis of 2,227 MRs, representing all changes made to the software in Department A from July 1997 to July 1999. The correlations among the variables are shown in the Table 1 (asterisks indicate log transformations).

The Table 2 shows the results of a multiple regression of all the predictor variables on interval (asterisks indicate log transformations).

The coefficients indicate that number of people, size, and diffusion significantly increase MR interval. The MR interval also increases with time and decreases with severity. Surprisingly, *given all other factors*, distributed MRs do not have significantly longer intervals than single-site MRs. A replication of this model using data from Department B also did not show an effect for distributed versus same-site MR intervals, given all the other factors in the model. We now turn to several hypotheses and a statistical modeling technique more able to capture all of the relationships among the important variables in order to better understand the full picture.

There are several possible explanations for our unexpected finding that the distributed versus same-site factor did not predict MR interval in our regression model:

- Large changes take longer to implement and also are more likely to involve multiple sites.
- Changes touching many modules take longer to implement and also are more likely to involve multiple sites.
- Distributed changes require participation of a larger number of people, which, in turn, introduces delay.

To investigate these hypotheses and to illustrate relationships among variables in our model we used graphical modeling techniques (see, e.g., [13], [40]). Fig. 1a shows the result of stepwise fitting of graphical Gaussian (otherwise known as covariance selection) models with threshold p-value of 0.0001. Again, the skewed variables were transformed. The resulting model contains only links that have high values of deletion deviance. Deletion deviance is the amount by which residual deviance increases if the link is removed from the model. Deletion deviances have approximately Chi-square distribution with one degree of freedom. The deletion deviances in the figure are very high, indicating very high significance of the relationships in the

TABLE 2
Results of a Multiple Regression of All the Predictor Variables

variable	coefficient	p-value
number of people*	0.35	< 0.001
diffusion*	0.43	< 0.001
size*	0.15	< 0.001
time	0.26	< 0.001
severity	-0.12	< 0.01
fix	-0.08	< 0.1
distributed	*	not significant

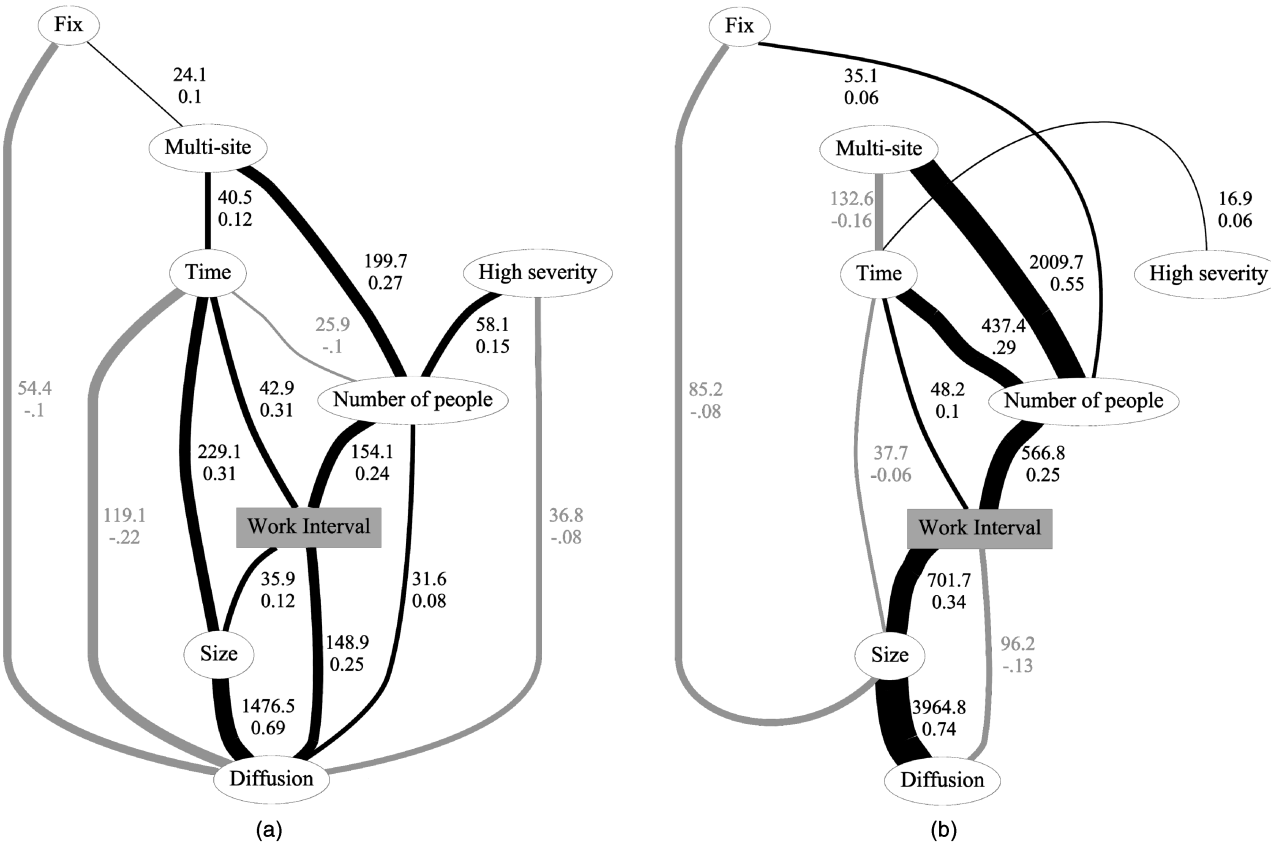

Fig. 1. Graphical model of MR interval for (a) Department A and (b) Department B.

model. The significance should be interpreted cautiously, of course, as in any technique involving model selection.

The model fit is determined by the difference of deviances between the fitted and saturated model. The absent links represent zero partial correlations, so there are 28 parameters representing all possible partial correlations in a saturated model between the eight variables we are considering. The saturated model, of course, has zero deviance, while the independence model (no links present) had a deviance of 3,697 with 28 degrees of freedom, and the model shown in Fig. 1a had a deviance of just 49 with 13 degrees of freedom (there are 15 links in the model). This indicates the model is a very good fit.

The nodes in Fig. 1 represent variables and links represent significant relationships among them. We use partial correlations, rather than sample correlations, to show the strength of these relationships between variables in the model. A partial correlation between variables X_1 and X_2 differs from a conventional sample correlation in that partial correlations show the strength of relationship between two variables *given* values of all other variables that are directly connected to X_1 or X_2 in the model (see definition, in, e.g., [40]). The black color indicates positive partial correlations, while gray indicates negative partial correlations. The thickness of a link shows its significance. The two numbers next to a link show the deletion deviance and the partial correlation. The variables that are not directly connected in the graph are independent given values of the other variables to which they have links in the model. We refer to such variables as not directly related.

Fig. 1 shows that the variables directly related to the MR interval, from most to least significant, are number of people, diffusion, and size. It also shows that compared to single-site MRs, distributed MRs tend to be associated with more people (and also have a slight tendency to increase in number over time and to be associated with bug fixes rather than new features).

Two of the plausible indirect relationships between the distributed character of work and MR interval would appear to be ruled out by this model. In particular, the hypothesis that *large changes take longer to implement and also are more likely to involve multiple sites* is not supported by the analysis. Similarly, the hypothesis that *changes touching many modules take longer to implement and also are more likely to involve multiple sites* receives no support. The multisite variable does not have a significant partial correlation with either size or diffusion.

The model is consistent, however, with the third hypothesis that *distributed changes require participation of a larger number of people, which, in turn, introduces delay.* Distributed changes are strongly related to the number of people who work on an MR, which, in turn, is strongly related to MR interval. It appears that splitting work across sites slows the work down primarily because it requires the involvement of more people than comparable work accomplished all at one site. To make sure that this trend is not simply caused by the fact that distributed MRs *must* involve at least two people (and single-site MRs occasionally involve only one person), we fitted a graphical model excluding single-person MRs. The resulting model still

TABLE 3
Correlations among the Variables for Department B

	interval*	number of people*	diffusion*	size*	time	severity	fix	distributed
interval*	1.00	0.35	0.20	0.38	0.17	0.00	0.01	0.16
number of people*		1.00	0.01	0.07	0.28	0.04	0.07	0.56
diffusion*			1.00	0.75	-0.01	0.00	-0.12	0.00
size*				1.00	-0.02	-0.03	-0.12	0.06
time					1.00	0.06	0.04	0.03
severity						1.00	0.02	-0.03
fix							1.00	0.05
distributed								1.00

indicated number of people as the most significant partial correlation for MR interval, and multiple sites as the most significant partial correlation for the number of people.

3.3 Replication

After this analysis for Department A was complete, we explored the generality of our findings by replicating the analysis on MR data from Department B. This department made a good comparison because it differed on many dimensions, as we mentioned in the previous section. Both shared some similarities, of course, since they resided in the same company. But, Department B involved different people, different sites, and a different type of software. Given these differences, the developers in the two departments tended to have rather different skills, different processes, and used different tools. Moreover, the distributed work in B involved coordination between a US and a European site, rather than among European and Asian sites. In order to see if the results are likely to hold up in a broad range of conditions, it was useful to have a department that differed in marked ways from our original data source.

The Table 3 shows the correlations among the variables for Department B, computed on the basis of 4,974 MRs, which represent all of the changes made to the software in Department B from July 1997 to July 1999 (as before, asterisks indicate log transformations).

A multiple regression of all variables on the interval again showed, as for project A, that after taking all other variables into account, whether the project was distributed did not have a significant effect on interval. (The full regression results contain no further insights, and are omitted in order to conserve space.) Fig. 1b shows the result of applying the same graphical modeling technique described in the previous section, i.e., a stepwise fitting of a graphical Gaussian model, using a threshold p-value of 0.0001. As before, the skewed variables were transformed. We chose MRs from the same range of dates as in the analysis of Department A MRs, to control for potential differences in business environment. The independence model had a deviance of 8,296 with 28 degrees of freedom while the fitted model had deviance of 79 with 16 degrees of freedom (there are 12 links in the model). With respect to the results of interest, the similarity to the results obtained with data from Department A is striking.

Fig. 1b shows that the variables directly related to MR interval, from most to least significant, are size, number of people, diffusion, and time. It also shows that the variable with the most significant relationship to the number people who work on the MR is whether the MR is distributed. The proportion of distributed MRs in this case tend to decrease over time. Also, and at first rather puzzling, MRs of comparable size tend to take less time if they are *more* diffuse. Further examination of the data shows that this effect comes primarily from very large MRs. This may indicate bottlenecks when only one part of the system with a limited number of experts needs very large modifications.

While there are several differences in the significant relationships in the Fig. 1 graphs, the relationships that bear on our hypotheses are essentially identical. As in Department A, two of the plausible indirect relationships between the distributed character of work and MR interval would appear to be ruled out by this model. In particular, the hypothesis that *large changes take longer to implement and also are more likely to involve multiple sites* is not supported by the analysis. Similarly, the hypothesis that *changes touching many modules take longer to implement and also are more likely to involve multiple sites*, receives no support. The multisite variable does not have a significant partial correlation with either size or diffusion.

As before, the model is consistent only with the third hypothesis, i.e., that *distributed changes require participation of a larger number of people, which, in turn, introduces delay*. Distributed changes are strongly related to the number of people who work on an MR, a variable that, in turn, is strongly related to MR interval. As in Department A, it appears that splitting work across sites slows the work down primarily because it requires the involvement of more people than comparable work accomplished all at one site. The similarity between the relevant parts of the graphical models indicates that the discovered relationships might be quite general and fundamental to distributed work.

Next, we examine differences between same site and distributed work in terms of social networks, communication, and coordination. These results provide additional understanding of the relationship between delay, number of people required to accomplish a change, and geographic distribution.

3.4 Social Networks: Distributed Communication, Cross-Site Attitudes

The mechanisms by which software engineers coordinate work may be rendered less effective as they operate across distance. In this section, we first describe several ways in

which we hypothesize that distributed social networks are less effective than same-site networks, both because of communication patterns and because of attitudes that distant workers have toward each other. We then present data from two administrations of a survey that bear on our hypotheses. Finally, we describe specific mechanisms by which less effective social networks can require involvement of more people in order to resolve an MR.

3.4.1 Hypotheses—Social Networks Less Effective

Distributed social networks may be less effective than local social networks in the following ways:

H1: Distributed social networks are much smaller than same-site social networks.

H2: There is much less frequent communication in distributed social networks compared to same-site social networks.

H3: People find it much more difficult to identify distant colleagues with needed expertise and to communicate effectively with them.

H4: People at different sites are less likely to have a common view of priorities than are people at the same site.

H5: People at different sites are less likely to perceive themselves as part of the same team than are people at the same site.

We now turn to an analysis of survey data that focus on these hypotheses. As noted above, all survey data came from Department A. We organize the results by their relevance to the hypotheses above.

3.4.2 Results from Survey

H1: Distributed social networks are much smaller than same-site social networks.

In order to get a rough estimate of the size of local and remote social networks, we asked people to

> (S2) Consider an average week. How many different people do you typically interact with at work during the course of the week from your <local, remote> site?[5] ($t = 12.4036$, $df = 77$, $p < .0001$).

(Questions labeled S1 appeared on the first survey, S2 appeared on the second.) As we expected, the results were quite different for local and remote sites. The mean for local was 16.0 and for remote 4.9. This difference is highly significant.

We also asked people to identify up to 10 people they communicate with at their local and remote site:

> Consider your most important current project. Select up to 10 coworkers involved with this project who are located at <your site, any other site>.

The wording of this question differs from the previous one—rather than asking about all of the people the respondent interacts with, we asked only about people working on the same project. We then counted up the names mentioned in order to get another comparison of the size of local and remote networks. The mean for local site is 7.6 and for remote site it is 4.8. As before, they report a significantly larger number of people at the local site than the distant site ($t = 7.3684$, $df = 91$, p-value $< .0001$).

These observations tend to support H1.

H2: There is much less frequent communication in distributed social networks compared to same-site social networks.

In the item where we asked our respondents to identify other people in local and remote sites they communicate with, we also asked them about the frequency of communication for each person. If we simply count up the number of local people and distant people, across all respondents, who fall into each communication frequency category, we produce the distribution shown in Fig. 2. The results of the two surveys are quite similar, showing much more frequent communication with local colleagues.

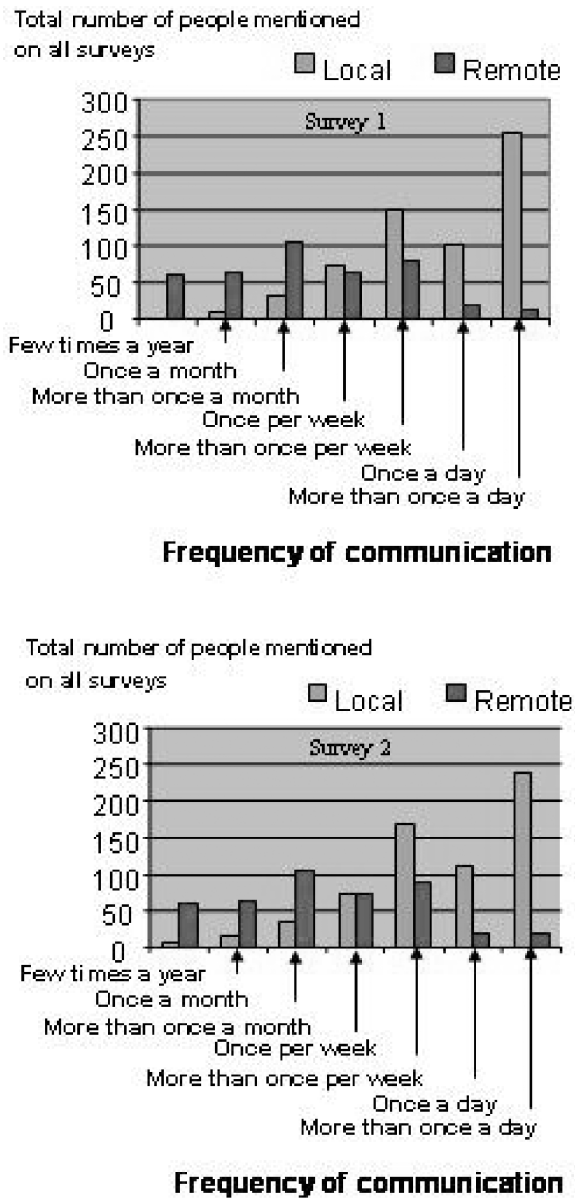

Fig. 2. Distribution of all people respondent communicates with, number of people by frequency of communication, surveys 1 and 2.

5. Again, because these data were bounded at zero, we performed the t-test on a square root transformation.

Respondents communicate with the majority of their local colleagues at least once a day. They communicate with the majority of colleagues at other sites, however, less that once a week. Communication with others at distant sites are much less frequent, and, of course, are much more likely to be conducted via e-mail, phone, and conference bridge, rather than face to face.

These results tend to support H2. The results described so far tend to indicate only that there are fewer opportunities to communicate with remote colleagues as compared to local colleagues. Next, we present results from several addition questions designed to try to identify some of the effects of this reduced level of communication.

H3: People find it much more difficult to identify distant colleagues with needed expertise and to communicate effectively with them.

We asked about the difficulty of identifying and finding people at local and distant sites:

> (S1) I lose time trying to figure out who to contact regarding my work. ($t = 4.44$, $df = 66$, $p < .0001$).

> (S1, S2) People I need to communicate with are difficult to find. (S1: $t = 2.82$, $df = 67$, $p = .006$; S2: $t = 2.59$, $df = 68$, $p = .01$).

There was a significant difference in responses for remote and local sites, with a much greater tendency to believe that people at remote sites were more difficult to find and to contact.

We also asked several questions designed to assess the extent to which important information flowed through the communication network, and about inadequacies and barriers:

> (S1) I often get useful work-related information through casual conversations ($t = 5.44$, $df = 64$, $p < .0001$).

> (S1) There have been times when I was accidentally excluded from information which was shared by my coworkers ($t = 3.56$, $df = 61$, $p = .0007$).

> (S2) My coworkers provide timely information about changes in current plans ($t = 6.3$, $df = 91$, $p < .0001$).

Again, the differences in responses across sites were highly significant and in the direction suggesting better communication within a single site. These results tend to support H3.

H4: People at different sites are less likely to have a common view of priorities than are people at the same site.

We asked questions in order to determine what kinds of misunderstandings about priority and scheduling of tasks that may have arisen locally and across sites. Interestingly, we found no evidence for greater cross-site misunderstanding. Questions probing these issues revealed that workers were no more likely to perceive disagreements about task priorities or about the clarity of task assignments among distant colleagues that for local colleagues:

> (S2) There is disagreement about task priorities. ($t = 1.09$, $df = 91$, not significant).

> (S2) When work is assigned, everyone is clear about his or her task. ($t = 1.62$, $df = 91$, not significant).

These results do not provide support for H4. There does not appear to be a perception that the priority of tasks is different across sites.

H5: People at different sites are less likely to perceive themselves as part of the same team than are people at the same site.

We asked several questions designed to discover whether developers at different sites felt as though they were really a team. Presumably, a feeling of common team membership can provide a strong bond that will motivate people to undertake tasks on behalf of other team members. We asked them to think about their most important current project as they answered the following questions:

> (S1, S2) I feel like I'm part of the same team as my coworkers. (S1: $t = 6.8833$, $df = 74$, p-value $< .0001$; S2:, $t = 6.0778$, $df = 91$, p-value $< .0001$).

> (S2) I feel accepted by my coworkers as a team member. ($t = 3.9402$, $df = 91$, p-value $< .0001$).

> (S2) My coworkers and I share the same team spirit. ($t = 5.9862$, $df = 91$, p-value $< .0001$).

The answers for all three questions tend to indicate substantially less of a team feeling among those at different sites. We also asked whether people feel motivated to solve problems in ways that benefit everyone:

> (S1) During meetings, my coworkers and I do our best to produce mutually beneficial solutions to problems. ($t = 2.3844$, $df = 70$, p-value $= 0.01$).

We also asked a general question designed to ascertain how readily people believed they could work with those from another site:

> (S1) My coworkers and I have work styles that fit well together. ($t = 3.1812$, $df = 71$, p-value $= 0.001$).

A related issue is how people respond to an overloaded colleague, i.e., are they willing to go "above and beyond" to help out when needed:

> (S2) I assist my coworkers with heavy workloads, beyond what I am required to do. ($t = 1.05$, $df = 91$, not significant).

The results show no significant difference in answers between local and remote. But, we also asked:

> (S2) My coworkers assist me with heavy workloads, beyond what they are required to do. ($t = 6.26$, $df = 91$, $p < .0001$).

For this question, the respondents reported a sizeable, highly significant difference between the help offered by their (more helpful) local and (less helpful) remote colleagues. The contrast in these two answers is quite intriguing. While people report that they assist local and remote colleagues equally, the same people report they are much more likely to receive assistance from local than remote colleagues.

The data support H5. There is substantial evidence that cross-site relationships, compared to same-site relationships, have less of a team orientation and are less oriented toward mutual benefit.

3.4.3 Less Effective Social Networks Require More People in Order to Perform the Work

In the previous section, we saw several specific ways in which cross-site social networks differ from local networks. There are several ways that these differences can lead to the participation of more people in distributed MRs. It is important to note that by "more people," we mean more people *who leave traces in the CM system*. This measure is taken from the MR data and the CM system only records participation of those who originate MRs or who contribute deltas. If people are involved only in other ways, e.g., giving advice, helping to locate experts, and so on, their participation is not recorded.

In the organizations under study, as we mentioned above, the initial assignment of work on an MR is made formally, by the change control board, to a particular individual we call the MR owner. In many cases, the MR owner requires the work of others in order to complete the MR. These further work assignments are in general handled informally. The MR owner locates people with the needed expertise and requests that they provide needed information or that they add the work to their own schedules. If the MR owner does not succeed in securing the needed assistance in this direct, informal way, he/she must escalate the request through the management chain.

Given this general way of recruiting technical resources for MRs, it is clear that the informal process relies on social networks. There are several ways in which the specific properties of cross-site social networks may lead to involvement of more people in resolving an MR. We describe six possible mechanisms below. In each case, we indicate the hypotheses that describe the specific properties of cross-site social networks that are involved in the mechanism. (There are other possible explanations, involving conflicting priorities, that we do not propose because H4 was not supported.)

- *H1, H2, H3.* Since it is more difficult across sites for an MR owner to identify and contact the person or persons whose expertise best matches the need, the MR owner will often recruit other people, each with imperfectly matched expertise. Thus, distributed MRs are likely to require more people in order to tap all the expertise needed to complete the MR.
- *H1, H2, H3.* Through the relatively constant communication with local colleagues, one is much more aware of the work going on locally and in a much better position to ask questions when the need arises. This makes it much more likely that one will avoid making changes that conflict with code written and maintained locally. It is harder to avoid creating conflicts with code developed and maintained at other sites. More conflicts are introduced, and additional people often get involved in order to resolve the conflicts.
- *H1, H2, H3.* It is more difficult for the MR owner to receive an informal consultation with an expert. MR owners, with some effectively rendered advice from an expert, may be able to quickly perform more of the work him/herself. (Recall that this type of consultation by the expert does not increase the number of people as we've measured it—people were counted only if they actually made changes.) When such consultations cannot be arranged, or if they do not convey sufficient information, the MR owner must ask the expert to add the work to his/her schedule and actually make the changes. Since this involves requesting and scheduling the expert's time, it is a slower process that also involves more people committing changes to an MR.
- *H1, H2, H3.* We assume that people who communicate more frequently are more likely to understand each other clearly. Given that assumption, it will be more difficult to communicate requests clearly across sites, often resulting in incomplete or incorrect changes. This requires additional people to complete the work.
- *H5.* If a request is made by an MR owner who is not really considered part of the "team," the person whose time is requested may be reluctant to place the request ahead of other scheduled work. Other, less qualified people who happen to be available, each of whom may have only part of the required expertise, are more likely to become involved in order to complete the MR.
- *H5.* Rather than refusing outright to help, distant staff may feel less motivated to respond to requests from other sites and may be less likely to fully and diligently complete the work. Participation of additional people is required to complete the work.

We view these explanations as plausible, given the support for the relevant hypotheses, but they are clearly speculative. They are best viewed as the bases for hypotheses to be tested in future work.

In this section, we have examined the properties of distributed social networks and plausible mechanisms that explain why they appear to function in a way that requires more people than same-site networks. For all of these mechanisms, we would expect that if sites are able to become more independent over time, the delay problems would be reduced. In the next section, we use change management data to look at trends in interdependence of sites.

3.5 Interdependence of Sites

Overall interdependence. Fig. 3 shows the interdependence of the work across sites, as measured by the proportion of distributed MRs to all MRs, plotted as a function of time. Since the work within an MR is tightly coupled, the proportion of distributed MRs can be used as a measure of the coupling of the work across sites. Overall, about 16 percent of the coding in Department A occurs in MRs that are split across sites. The remaining 84 percent occurs in single-site MRs. The work in Department B is much more tightly coupled across sites, varying from 24 percent to 45 percent of MRs being distributed.

As Fig. 3 shows, Department A appeared to increase substantially in interdependence over time, as the proportion of distributed changes increased from about 12 percent

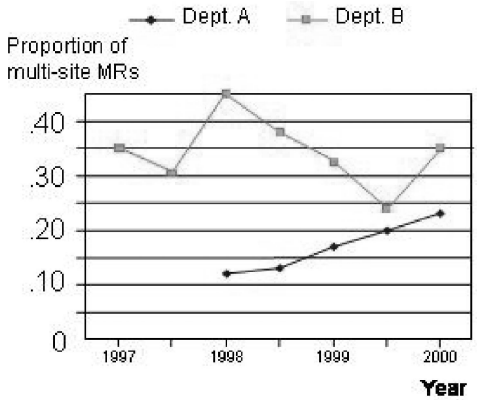

Fig. 3. Proportion of multiple-site MRs over time.

to about 23 percent. The fractions of distributed changes significantly increased over time ($p < 0.001$ using logistic regression with predictor time and response indicator of whether the MR is distributed across locations). Department B shows a somewhat different picture, with a higher overall level of interdependence but no clear linear trend.

The most important point is that it does not appear that either department reduced the interdependence among sites in any consistent way. The interdependence remains at high levels over the periods for which we have data, i.e., 2.5 years for Department A and 3.5 years for Department B. Cross-site communication and coordination issues appear to be enduring problems in both departments.

4 Conclusions

In this section, we describe how our results bear on our original research questions, and we suggest specific tools and practices that hold potential for addressing the problems we identified. It is important to keep in mind that these results might differ for organizations that distribute work across sites in different ways (e.g., [18]), or that use other business arrangements, such as outsourcing. The possible effects of such arrangements are open questions for future research.

4.1 Addressing Our Research Questions

Does distributed work introduce delay, as compared to same-site work? The answer would appear to be "yes, but indirectly." Both the survey data and the MR data from two organizations indicate that work that spans sites takes longer than work that does not cross sites. The fact that all sources of data indicate substantial cross-site delays increases our confidence in this result. Two independent analyses of MR data, taken from different organizations, with different people, different locations, and building very different kinds of software, show remarkable similarity in all relevant respects. In particular, they show that distributed MRs require an interval of about 2.5 times longer than an MR where all the work is at a single site. Survey data, in which developers are asked about the relative length of same-site and cross-site delays, produces yet another confirmation of this result.

What factors influence the time interval required to make a software change? What role, if any, does spreading work across multiple sites play in lengthening this interval? Graphical models showed that size, diffusion, and number of people were all directly related to MR interval, in data taken from both departments. In both cases, there was no direct link between MR interval and the distributed character of the work.

The results from both MR analyses have clear implications about mechanisms that could produce the delay. The much longer interval for distributed work seems clearly not to be a side effect of distributed changes being either larger (number of deltas) or more complex (number of modules touched). Rather, more people tend to be involved in distributed work than in comparable same-site work, and the involvement of more people has a powerful influence on the MR interval. This suggests that distributed communication, coordination, and/or social networks may differ from their single-site counterparts in a way that requires more people to participate, thereby introducing delay.

What differences are there between same-site and distributed social "networks" and their effectiveness? We identified several differences, including very large differences in the size of the communication network and the frequency of communication. These characteristics apparently led to a very restricted flow of information across sites, creating greater difficulty in finding people and reducing the likelihood of obtaining useful information from them. The nearly complete lack of informal, "water cooler" conversation appears to have the consequence that people know much less about what distant colleagues are doing, who has expertise in what area, what the current status of plans is, and so on. In general, it seems that there is relatively little understanding of the overall "context" or background information at distant sites.

There is substantial evidence that distant colleagues feel less "teamness," experience some conflict in work styles and report that their distant colleagues are less likely to help out when workloads are especially heavy. The lack of "teamness" presumably reflects the fact that distance interfered with the usual stages by which individuals become coherent groups or teams. McGrath [29], for example, speculated that teams forced to communicate primarily through technology, rather than face to face, will have a more difficult time with the "middle" stages of a task, i.e., stages involving solutions for technical issues and resolving conflicts among people. We often heard developers mention, in particular, that both complex technical issues and contention presented difficulties when they arose with distant colleagues.

While it is plausible that these characteristics are a result of the differences between same-site and distributed communication, there are many other factors that could also contribute to this. In particular, the respondents in different locations have different cultural backgrounds, different work histories, are sometimes native speakers of different languages, and are likely to have somewhat different training. The evidence presented here does not allow us to determine the cause of the lack of "teamness," but one hypothesis for future work is that the attenuated

communication across sites contributes to this. One could rule out many of the cultural factors, for example, by examining teams distributed among sites within the same country, by examining relationships of those who have expatriot assignments with their local colleagues, and by following distributed groups over time as cross-site communication is enhanced by introducing new tools and practices.

These differences between same-site and distributed social networks could, in several specific ways we identified, force the involvement of more people in a distributed MR.

To what degree is work at the different sites interdependent? Does interdependence diminish over time? We use the proportion of MRs that involved people at more than one site as a straightforward measure of interdependence. In the departments we observed, the change data showed an increasing trend toward site interdependence in Department A and no clear linear trend in Department B. This runs counter to some observations in the literature (e.g., [36]), where distributed work was spontaneously rearranged to make it less closely coupled. This may reflect constraints about where the relevant expertise is, the addition of new people and sites that must be given some of the work even if the coupling to other work is tighter than one would like, or it may reflect a tendency of a software architecture to deteriorate over time so as to limit the ability to isolate changes to particular parts of the product [14].

Regardless of the reasons, in the organizations in this study, high levels of interdependence were maintained for the entire 2.5 - 3.5 years, showing no consistent reduction. This suggests that reducing interdependence, when it is possible at all, will require the application of some novel strategy. It also suggests that distributed work is a chronic condition. Substantial investment in tools, practices, and perhaps other types of solutions may be justified if the organization desires faster cycle times.

In this section, we have discussed how the data we presented addressed the research questions we posed at the outset. In the next section, we draw out the implications of our findings for the design of tools for distributed work.

4.2 Speeding Global Collaboration

There are a number of hints in the data about the kinds of practices and technologies that might be effective in reducing MR interval. In this section, we discuss two primary strategies one might pursue. The first is to avoid the problem. Ideally, one might locate all development resources at one site. But, this will often not be an option simply because of where experts are located, because various market requirements make it impossible, or because there is insufficient technical staff available. Assuming that locating all development at one site is not an option, there are ways to decouple the work at the different sites so that they can operate more independently from one another. The second strategy is to use practices and technology to facilitate communication across sites in order to approximate the properties of same-site social networks in distributed social networks. Presumably, one would focus on goals such as facilitating frequent informal communication, helping to identify experts, and making it easier to initiate effective communication with a distant colleague.

4.2.1 Optimally Splitting Work across Sites

There are a number of ways that a software organization can split work across sites [18]. For example, some organizations hand off work to other sites after completing certain process steps, e.g., handing off a design to another site for coding, or handing off a partially tested product for further stages of testing. Other organizations may develop a core product at a central site and customize it locally for various markets. Yet another strategy divides the product according to its structure and develops various product parts at different sites, resulting in an organizational structure that mirrors the product structure [21], [8], or product features [12]. These different strategies can be expected to be more or less effective, depending on a variety of circumstances [18].

For those many organizations that split the work along the lines of product structure, considerations of modular design are extremely important [38], in order to isolate the effects of changes. It is possible to use the change history to minimize the interdependence of work at the various sites. Mockus and Weiss [32] have developed an algorithm for selecting parts of the product that could be most easily moved. A similar algorithm could be used to compare several alternative distributions of the product proposed on other grounds.

4.2.2 Increasing Communication

As we pointed out in the introduction, there have been many studies of the behavior of developers (e.g., [39], [22]), and all of them tend to show that much time is spent in communication, both in relatively "formal" scheduled meetings, as well as in hallways and the offices of colleagues. As long as people are colocated, it seems that people's natural gregarious tendencies can be relied upon to disseminate information about expertise, context, status, workloads, urgency, availability, and a host of other subtleties that are critical to effective communication and coordination. When teams and organizations are split apart, much of this is lost, to their great detriment.

There are many tools that could be used to restore this communication channel. The organizations described in the study made much use of telephone and e-mail, but this seemed insufficient, as the data showed. Additional forms of text-based communication, such as instant messaging [34] and MUD rooms [7] have proven to be an effective means of communication in scientific and business settings. We introduced an instant messaging and group chat tool in both of the departments studied here. The tool was adopted and regularly used by some groups, and not at all by others. An analysis of the content of group chat indicates that it was primarily used for work-related, rather than purely social, conversation [19]. It also tends to be much less intrusive than a telephone since receiving an instant message does not create such a jarring interruption, and answering can be deferred to a convenient time.

While some groups warmed quickly to group chat and instant messaging, many others did not [20]. People enjoy chatting in the hall, but it doesn't seem like "work," so, if the opportunity is taken away by distance, there does not

seem to be a compelling and legitimate reason to find a substitute. Moreover, the alternatives are admittedly somewhat awkward. It takes a while before a newcomer is comfortable "chatting" with an instant messaging tool. This combination of factors, i.e., the lack of perceived need and the initial awkwardness, can be a powerful deterrent.

In addition to communication tools, there are practices that have proven effective in increasing communication. For example, sending individuals to other sites in the role of liaison has often proven effective, e.g., [4].

4.2.3 Finding Experts

In general, developers have difficulty finding the experts they need (e.g., [2]) and, as we found in the current study, the problem is greater when the needed expertise is at a distant site. There is relatively sparse communication across sites and correspondingly few opportunities to learn who does what, who has what expertise, and to be aware of where they are now. There is relatively little opportunity to learn in depth about the expertise of those at other sites.

It is possible to obtain descriptions of each person's expertise, then make them available in some way. In our experience, this is relatively ineffective because such descriptions are very difficult to construct in a common, meaningful vocabulary and because they are very difficult to maintain. We adopted a different strategy, using change history data to identify who has worked on what kinds of things, hence, who has the needed expertise. We developed and deployed a tool that uses this strategy called Experience Browser [30], which provides a visualization of the CM system and is designed to make it easy to find and contact an appropriate expert.

4.2.4 Awareness

Instant messaging applications have a secondary function, in addition to allowing the exchange of messages. They give the user an indication of whether a person is available. One can discover, for example, that a particular user is logged on to the server. Often one can also see how long the person's machine has been idle, which may indicate such things as whether the person is in a meeting.

Shared calendars can also play a central role by letting people at other sites know where someone is, when they might be free and even who they have been meeting with [37]. We have had a similar experience, deploying a Web-based calendar tool in the development organization. It is receiving heavy use, i.e., thousands of hits per month.

Judicious use of presence awareness technology has the potential to substantially lower the difficulty and frustration associated with contacting a remote colleague. One can often time one's communication, for example, for a moment when someone is available to reply. Someone's "arrival" online may also serve as a memory trigger, reminding the user of a deferred conversation one ought to have with the recently "arrived" colleague, as often happens in hallway conversation.

5 Contribution

We presented an analysis of modification request data showing a relationship between distributed development and the number of people involved, as well as a relationship between the number of people and the MR interval. A replication of the analysis on data from another organization showed these same relationships. We believe this may reveal a fundamental relationship between distributed work, social networks, and delay. Our survey responses have helped to pin down the specific ways in which distributed networks are less effective than same-site networks. We have also speculated on several ways in which these specific deficiencies could lead to delay, which paves the way for further research in this area.

We have also provided an example of empirical research that combines several sources of data for cross-validation and to further explore issues raised by the initial studies. We show the practical importance of the findings for collaborative development by describing implications for tools and practices directed toward the specific problems we identified.

We believe that a deep understanding of the issues of global development is critical for the success of today's far-flung software organizations. We should not forget, however, that distances far short of those studied here have a profound effect on communication [3], [27]. In general, people 30 meters apart communicate no more frequently than those separated by oceans. One can only suspect that, if significant progress can be made on the problems of global development, many of those solutions can be applied to greatly improving development spanning a campus, different floors, or even a long corridor. We believe the goal of collaborative software development should be to create for the team a "virtual 30 meters"—to make it as simple, natural, and straightforward to stay in contact and to collaborate with one's team, wherever located, as it is now for colleagues occupying adjacent offices.

Acknowledgments

The authors would like thank their colleagues Thomas Finholt, Rebecca Grinter, Erik Hofer, Nishant Jain, Gary Olson, Dewayne Perry, Elena Rocco, and David Weiss. They also thank our development partners, Roger Levy, Daniel Owens, Frank Wales, and Michael Vander Wielen, who have provided support and focus. This work was done while the authors were at Bell Laboratories, Lucent Technologies.

References

[1] M.J. Abel, "Experiences in an Exploratory Distributed Organization," *Intellectual Teamwork: Social Foundations of Cooperative Work,* J. Galegher, R.E. Kraut, and C. Egido, eds., pp. 489-510, 1990.
[2] M.S. Ackerman and C. Halverson, "Considering an Organization's Memory," *Proc. Computer Supported Cooperative Work Conf.,* pp. 39-48, 1998.
[3] T.J. Allen, *Managing the Flow of Technology.* MIT Press, 1977.
[4] R.D. Battin, R. Crocker, J. Kreidler, and K. Subramanian, "Leveraging Resources in Global Software Development," *IEEE Software,* vol. 18, no. 2, pp. 70-77, Mar./Apr. 2001.
[5] F.P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering.* Addison Wesley, 1995.
[6] E. Carmel, *Global Software Teams.* Prentice-Hall, 1999.
[7] E.F. Churchill and S. Bly, "It's All In the Words: Supporting Work Activities With Lightweight Tools," *Proc. Int'l ACM SIGROUP Conf. Supporting Group Work,* pp. 40-49, 1999.
[8] M.E. Conway, "How Do Committees Invent?" *Datamation,* vol. 14, no. 4, pp. 28-31, 1968.

[9] B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process for Large Systems," *Comm. ACM*, vol. 31, no. 11, pp. 1268-1287, 1988.
[10] M.A. Cusumano and D.B. Yoffie, *Competing on Internet Time.* The Free Press, 1998.
[11] P. Dourish and S. Bly, "Portholes: Supporting Awareness in a Distributed Work Group," *Proc. ACM Conf. Human Factors in Computing Systems*, pp. 541-547, 1992.
[12] C. Ebert and P. DeNeve, "Surviving Global Software Development," *IEEE Software*, vol. 18, no. 2, pp. 62-69, Mar./Apr. 2001.
[13] D. Edwards, *Introduction to Graphical Modeling*, second ed. Springer Verlag, 2000.
[14] S.G. Eick, T.L. Graves, A.F. Karr, J.S. Marron, and A. Mockus, "Does Code Decay? Assessing the Evidence from Change Management Data," *IEEE Trans. Software Eng.*, vol. 27, no. 1, pp. 1-12, Jan./Feb. 2001.
[15] R.S. Fish, R.E. Kraut, and R.W. Root, "Evaluating Video as a Technology for Informal Communication," *Proc. ACM Conf. Human Factors in Computing Systems*, pp. 37-48, 1992.
[16] J. Galbraith, *Organizational Design.* Addison-Wesley, 1977.
[17] R.E. Grinter, "Recomposition: Putting It All Back Together Again," *Proc. ACM Conf. Computer Supported Cooperative Work*, 1998.
[18] R.E. Grinter, J.D. Herbsleb, and D.E. Perry, "The Geography of Coordination: Dealing with Distance in R & D Work," *Proc. Int'l ACM SIGROUP Conf. Supporting Group Work*, pp. 306-315, 1999.
[19] M. Handel and J.D. Herbsleb, "What Is Chat Doing in the Workplace?" *Proc. ACM Conf. Computer Supported Cooperative Work*, pp. 1-10, 2002.
[20] J.D. Herbsleb, D.L. Atkins, D.G. Boyer, M. Handel, and T.A. Finholt, "Introducing Instant Messaging and Chat into the Workplace," *Proc. ACM Conf. Computer-Human Interaction*, pp. 171-178, 2002.
[21] J.D. Herbsleb and R.E. Grinter, "Splitting the Organization and Integrating the Code: Conway's Law Revisited," *Proc. Int'l Conf. Software Eng.*, pp. 85-95, 1999.
[22] J.D. Herbsleb, H. Klein, G.M. Olson, H. Brunner, J.S. Olson, and J. Harding, "Object-Oriented Analysis and Design in Software Project Teams," *Human-Computer Interaction*, vol. 10, nos. 2-3, pp. 249-292, 1995.
[23] J.D. Herbsleb, A. Mockus, T.A. Finholt, and R.E. Grinter, "An Empirical Study of Global Software Development: Distance and Speed," *Proc. Int'l Conf. Software Eng.*, pp. 81-90, 2001.
[24] "Global Software Development," *IEEE Software*, J.D. Herbsleb and D. Moitra, eds., vol. 18, no. 2, Mar./Apr. 2001.
[25] D. Hindus, M.S. Ackerman, S. Mainwaring, and B. Starr, "Thunderwire: A Field Study of an Audio-Only Media Space," *Proc. Computer Supported Cooperative Work Conf.*, pp. 238-247, 1996.
[26] E. Isaacs, A. Walendowski, and D. Ranganathan, "Hubbub: A Sound-Enhanced Mobile Instant Messenger that Supports Awareness and Opportunistic Interactions," *Proc. ACM Conf. Human Factors in Computing Systems*, pp. 179-188, 2002.
[27] R.E. Kraut, C. Egido, and J. Galegher, "Patterns of Contact and Communication in Scientific Research Collaboration," *Intellectual Teamwork: Social Foundations of Cooperative Work*, J. Galegher, R.E. Kraut, and C. Egido, eds., pp. 149-171, 1990.
[28] R.E. Kraut and L.A. Streeter, "Coordination in Software Development," *Comm. the ACM*, vol. 38, no. 3, pp. 69-81, 1995.
[29] J.E. McGrath, "Time Matters in Groups," *Intellectual Teamwork: Social Foundations of Cooperative Work*, J. Galegher, R.E. Kraut, and C. Egido, eds., pp. 23-61, 1990.
[30] A. Mockus and J.D. Herbsleb, "Expertise Browser: A Quantitative Approach to Identifying Expertise," *Proc. Int'l Conf. Software Eng.*, pp. 503-512, 2002.
[31] A. Mockus and L.G. Votta, "Identifying Reasons for Software Change Using Historic Databases," *Proc. Int'l Conf. Software Maintenance*, pp. 120-130, 2000.
[32] A. Mockus and D.M. Weiss, "Globalization by Chunking: A Quantitative Approach," *IEEE Software*, vol. 18, no. 2, pp. 30-37, Mar./Apr. 2001.
[33] A. Mockus and D.M. Weiss, "Predicting Risk of Software Changes," *Bell Labs Technical J.* vol. 5, no. 2, pp. 169-180, 2000.
[34] B.A. Nardi, S. Whittaker, and E. Bradner, "Interaction and Outeraction: Instant Messaging in Action," *Proc. ACM Conf. Computer Supported Cooperative Work*, pp. 79-88, 2000.
[35] A. Obata and K. Sasaki, "OfficeWalker: A Virtual Visiting System Based on Proxemics," *Proc. ACM Conf. Computer Supported Cooperative Work*, pp. 1-10, 1998.
[36] J.S. Olson and S. Teasley, "Groupware in the Wild: Lessons Learned from a Year of Virtual Collocation," *Proc. ACM Conf. Computer Supported Cooperative Work*, pp. 419-427, 1996.
[37] L. Palen, "Social, Individual, and Technological Issues for Groupware Calendar Systems," *Proc. ACM Conf. Human Factors in Computing Systems*, pp. 17-24, 1999.
[38] D.L. Parnas, "On the Criteria to Be Used in Decomposing Systems into Modules," *Comm. the ACM*, vol. 15, no. 12, pp. 1053-1058, 1972.
[39] D.E. Perry, N.A. Staudenmayer, and L.G. Votta, "People, Organizations, and Process Improvement," *IEEE Software*, vol. 11, no. 4, pp. 36-45, 1994.
[40] J. Whittaker, *Graphical Models in Applied Multivariate Statistics.* Wiley, 1990.

James D. Herbsleb received the MS degree in computer science from the University of Michigan, Ann Arbor, and the PhD degree in psychology from University of Nebraska, Lincoln. He has held positions as a postdoctoral research fellow at the University of Michigan, and a senior member of the technical staff of the Software Engineering Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania. He is an associate professor in the School of Computer Science at Carnegie Mellon University. Previously, he was a member of the Software Production Research Department, Bell Laboratories, Lucent Technologies. He led the Bell Labs Collaboratory project, conducting empirical studies of geographically distributed software development and designing collaborative applications and services. For the past 12 years, he has conducted research and published in the areas of collaborative and open source software engineering, human-computer interaction, and computer supported cooperative work. For the last five years, his work has focused on collaboration technology and processes to support large globally distributed software development projects.

Audris Mockus received the BS and MS degrees in applied mathematics and system programming from the Moscow Institute of Physics and Technology in 1988. In 1991, he received MS degree and, in 1994, he received the PhD degree in statistics from Carnegie Mellon University. He conducts research of complex dynamic systems. He designs data mining methods to summarize and augment the system evolution data, interactive visualization techniques to inspect, present, and control the systems, and statistical models and optimization techniques to understand the systems. He works in Software Technology Research Department at Avaya Labs. Previously, he worked in Software Production Research Department at Bell Labs.

▷ **For more information on this or any computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.